

MPEG layered stream의 상관 관계를 이용한 저장 기술

이 연구는 BK21 충남대학교 정보통신인력양성사업단의 지원을 받았음

김현정*, 이흥기, 김현정, 김상형, 이성인, 유관종,
충남대학교 컴퓨터과학과

Storage policy using dependency relation of MPEG layered stream

Hyun-Jeong Kim, Heung-Ki Lee, Hyun-Jeong Kim,
Sang-Hyoung Kim, Sung-In Lee, Kawn-Jong Yoo
Dept. of Computer Science, Chung-nam Univ.

E-mail : {ppuyo,helius,hjkim,shkim}@cs.cnu.ac.kr, silee@health.ac.kr, kjyoo@cnu.ac.kr

요 약

대용량의 MPEG 비디오 데이터를 시간적 개념의 layer coding 기법과 화질적 개념의 layer coding 기법을 통해서 나누는 방법을 통합하여 적용 시킴으로써, 세분화되어진 layered stream을 얻을 수 있다. 그러나, 이러한 layered stream을 어떠한 정책을 사용하여 저장을 수행하는 것이 효율적인지에 대해서는 현재로서는 연구가 미흡한 편이다. 이에 layered stream의 관계를 이용하여 load-balance을 구성하는 각 요소의 상관 관계를 고려하여 병렬 VOD서버를 구성하는 각 노드마다 load balancing을 이루도록 데이터를 저장하는 방법에 대해 제안하고자 한다.

1. 서론

네트워크 환경의 발전으로 인하여 인터넷 사용 인구가 증가하고 있으며, 이를 통해 다양하고 편리한 서비스를 제공받을 수 있게 되었다. 기존의 단순한 텍스트의 서비스뿐만 아니라, 오디오와 비디오 등 멀티미디어 데이터 서비스를 받을 수 있게 되었다. 그러나 네트워크환경이 빠르게 발전되고 있지만, 수 많은 인터넷 사용자들의 멀티미디어 서비스를 모두 충족시켜 주는 것은 현실적으로 불가능 하다. 더군다나

이질적인 환경의 인터넷이나 대역폭의 변화가 심한 모바일(mobile) 네트워크에서는 사용자가 만족할 만한 멀티미디어 서비스를 제공받기가 쉽지 않다.

전송량이 큰 멀티미디어 데이터를 서비스를 위해 일정량의 대역폭이 보장되어지기 힘든 현재의 인터넷환경에서, 사용자들이 만족할 만한 멀티미디어 서비스를 제공받기는 힘든 일이다.

현재의 인터넷 환경에서 가능한 멀티미디어 서비스를 제공하기 위하여 많은 연구가 진행 중이다. 이러한 연구 중의 하나가 MPEG 비디오 데이터를 여

러 개의 layered stream으로 나누는 것이다. 그리고, 전송하려는 상황에 맞추어서 layered stream을 선택적으로 전송하는 것이다. 데이터양이 많은 멀티미디어의 비디오 데이터를 계층(Layer) 개념을 이용하여 나누어 네트워크의 사정에 맞게 전송하는 것이다.

또한, 사용자가 만족할 만한 멀티미디어 서비스를 제공하지 못하고 있는 이유중의 하나는 멀티미디어 서버의 시스템의 속도 때문이다. 멀티미디어 파일과 같은 큰 파일의 처리 방법은 서버의 시스템 향상에 있어서 중요한 문제이다. 이런 문제점의 해결방법으로 병렬 VOD서버들이 나오고 있다.

본 논문에서는 기존에 연구되었던 MPEG 멀티미디어 데이터를 계층적으로 나누는 개념을 사용하여 데이터를 나눈 후에 여러 노드의 balancing을 유지하면서 저장하는 방법에 대하여 언급한다. 즉, 네트워크와 환경에 따라서 전송되어지는 stream의 개수를 정하게 된다. 즉, 가변적인 네트워크 환경에 맞추어서 전송되어지는 stream의 수와 크기를 유동적으로 변경시키는 것이다.

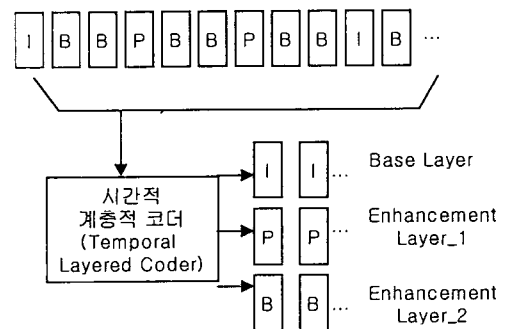
본 논문의 2장에서는 MPEG 데이터를 시간적 개념과 화질적 개념의 계층으로 나누는 2가지 방법에 대하여 살펴보고 3장에서는 2가지 개념을 모두 이용하여 멀티미디어 파일 서버를 더욱 효과적으로 사용할 수 있는 저장방법에 대하여 연구한다. 그리고 4장에서는 본 논문에서 제시한 방법에 대한 고찰을 마지막으로 5장에서는 결론을 맺는다.

2. 관련연구

2.1 Temporal Scaling 기법 [1]

Temporal Scaling 기법은 MPEG 비디오 stream의 픽처 층이 I 픽처, P 픽처, B 픽처로 구분 되어있는 것에 착안하여 개발된 Temporal Layered coding 기법이다. 이 방식은 전체 프레임에서 일부분의 프레임을 제거 함으로써, 전송되어지는 크기를 조절하는 계층적 코딩 기법이다.

I 픽처와 시퀀스 층과 GOP(Group of Picture)층을 재생에 필요한 최소한의 계층인 기본계층 (Base layer)으로 정의하고 고위 계층 (Enhancement layer) 으로는 P 픽처와 B픽처를 각각 고위계층 1 (Enhancement Layer_1) 과 고위계층 2 (Enhancement Layer_2) 로 정한다

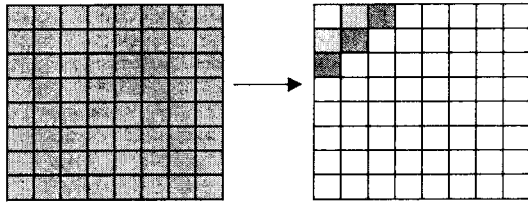


[그림1] 시간적 계층적 코딩의 개념도

위에서 정의한 3개의 계층을 이용하여 통신망 환경에 따라 MPEG 비디오 스트림을 서비스 해주게 된다. 대역폭이 충분한 통신망에서는 기본 계층과 고위 계층-1,2를 함께 전송 하고 대역폭이 충분하지 못한 통신망에서는 기본 계층만을 서비스 하는 식으로 서비스 하고자 하는 시점의 네트워크 환경에 맞추어 전송 되어지는 stream의 수를 결정하게 된다.

2.2 Fidelity Scaling 기법 [2]

Fidelity Scaling 기법은 MPEG 비디오의 구성 단계 중 최하위 층인 블록(Block) 층의 계수들을 이용하여 계층화하는 기법이다. DCT를 수행하는 단위는 8 X 8의 Block인데 이 DCT Block 을 5개의 계층으로 계층화 하는 것이다.[그림2]



DCT Block

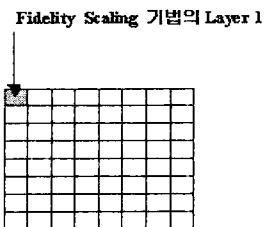
[그림2] Fidelity Scaling 의 개념

DCT블록의 계수들은 좌측 최상위의 블록이 가장 높은 계수치 값을 갖고 그 주변으로 점차 작은 값을 갖는 분포를 갖게 되는데 이러한 분포에 착안하여 Fidelity Scaling 기법에서는 DC를 기본계층으로 하고 AC1~2를 고위계층 1, AC3~5를 고위계층 2, AC6~9를 고위계층 3, AC10~AC63을 고위계층 4로 계층화 한다.

앞에서 언급했듯이 두 가지 방식을 동시에 하나의 stream에 적용을 하면 [표 1]과 같은 총 15개의 Layer를 생성할 수가 있다. 이러한 방식으로 생성된 stream을 Layer 1에서 15까지 네트워크환경에 따라 전송 여부를 결정한다.

T1S1	Temporal 1 / Fidelity(Spatial) 1 Layer
T1S2	Temporal 1 / Fidelity(Spatial) 2 Layer
...	...
T3S4	Temporal 3 / Fidelity(Spatial) 4 Layer
T3S5	Temporal 3 / Fidelity(Spatial) 5 Layer

[표 1]Temporal/Fidelity(Spatial) layer 적용 예



I picture (Temporal Scaling기법의 Layer 1)

[그림 3]Temporal/Fidelity Scaling방법의 점목

3. Layered stream의 저장 기법

3.1 Stream 단위의 저장

QoS에 따라 전송되는 데이터의 양을 조절하기 위해서는 나누어진 stream 의 임의의 위치에 접근하여야 한다. 시스템의 성능을 위해서, 나누어진 stream 단위로 데이터를 분산 저장한다.

3.2 Parallel VOD (Video On Demand)

Layered coding 기법을 적용함으로써, 총 15개의 layered stream을 생성할 수 있게 되었고, 이를 통해 나누어진 stream들을 어떠한 정책을 사용하여 배치할 것인가 하는 문제가 남았다. 단일 노드로 구성되어 있는 시스템에서는 layered stream의 배치는 별 문제가 되지 않는다. 왜냐하면, layering 된 데이터를 마치 하나의 stream인 것처럼 저장하였다가, 네트워크 상황에 맞게 사용자에게 전송하기만 하면 되기 때문이다. 그러나 이러한 시스템 구성은 서버의 load가 커지게 되고 이를 수용하기 위해서는, 서버 system의 구성에, 많은 비용이 들어가게 된다는 문제점이 있다.

이러한 문제점을 보완하기 위해서, 현재 많은 연구가 이루어지고 있는 분야가 바로 Parallel VOD(Video On Demand)이다. 리눅스를 이용하여 여러 대의 서버를 연결하여 마치 하나의 고성능 서버가 동작하는 것처럼 보이게 하는 것이다. 이러한 parallel VOD 서버에 layered stream을 어떻게 배치할 것인가를 결정하는 것이다. 일정 알고리즘을 통하여 stream을 배치시킴으로써, parallel VOD 서버를 구성하는 여러 개의 노드 중에서 특정의 노드에 사용자 요청이 집중되지 않도록 부하를 분산시킨다.

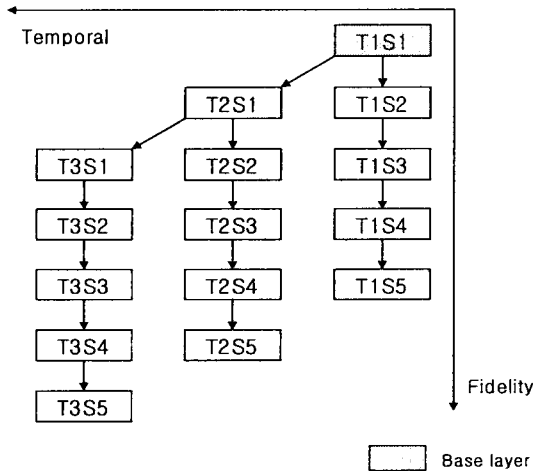
3.3 Hierarchy Structure

15개의 stream으로 분리되어진 layered stream은 각기 독립적으로 존재하는 stream이지만 재생을 위

해서는 서로간의 상관 관계를 따라야 한다. 고위 계층 stream이 부호화되기 위해서는 기본 계층과 특정 layer의 하위 layer가 존재해야만 가능하다. 따라서, 본 논문은 다음과 같은 rule을 제안한다.

1. Stream에 대해서 재생 하기 위해서는 기본 계층인 T_1S_1 레벨은 반드시 필요하다.
2. T_nS_1 ($1 \leq n \leq 3$) layer의 데이터를 재생하기 위해서는 $T_{n-1}S_1$ 레벨의 데이터가 필요하다.
3. T_nS_m ($1 \leq n \leq 3, 1 \leq m \leq 5$) layer를 재생하기 위해서는 T_nS_{m-1} layer가 필요하다.

이러한 dependence rule을 바탕으로 각각의 layered stream의 dependency 관계를 tree로 나타내면 [그림 4]와 같다.



[그림 4] Dependency tree

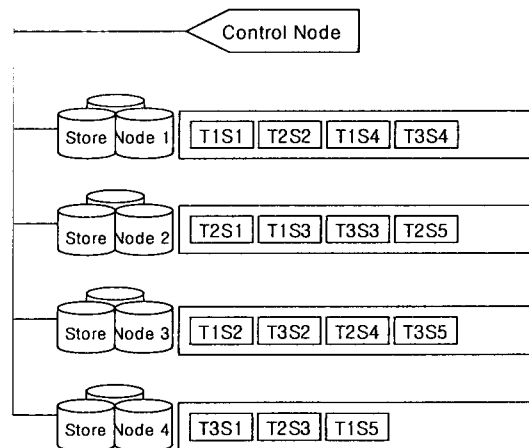
위의 dependency tree에서 보는 바와 같이, 기본 계층은 network의 QoS에는 관계없이 반드시 필요하게 된다. 따라서, 이 layered stream인 T_1S_1 stream을 배치한 노드에 대해서는 항상 접근이 이루어진다. 전송 layer를 결정하는 알고리즘에 따라 유동적이지만, T_2S_1 과 T_1S_2 stream이 기본 계층을 제외하고는 가장 많은 접근율을 보인다고 할 수 있다. 결국, T_1S_1 stream에 가까운 쪽의 stream이 저장된 노드일 수록 많은 접근이 이루어진다고 할 수 있다. 그러면

이러한 접근 성질에 따라 각각의 stream에 대한 우선 순위를 정하면 [표 2]와 같아진다.

1 T1S1	2 T1S2	3 T1S3	4 T1S4	5 T1S5
2 T2S1	3 T2S2	4 T2S3	5 T2S4	6 T2S5
3 T3S1	4 T3S2	5 T3S3	6 T3S4	7 T3S5
... : Dependency				
TnSm: Layered Stream				

[표 2] Layered Stream별 중요도

이러한 중요도표를 적용하여, 중요도가 높은 stream을 노드의 부하를 고려하여 시스템에 분산 배치 시킨다. 즉, 사용자의 요청이 하나의 노드에 집중되어지는 현상을 막는 것이다. 이것을 round-robin방식을 적용하여 parallel VOD 서버의 각 노드에 배치하도록 하면, [그림 5]와 같은 형태로 배치되어질 수 있다.



[그림 5] Layering example

4. 고찰

이번 논문에서는 15개의 layered stream을 하나의 parallel VOD서버에 효과적으로 배치하는 방법에 대해서 이야기를 하였다. 그러나, 아직 고려되어지지 못한 많은 문제점이 많이 남아 있다. 우선, 각각의 layered stream의 크기가 layer별로 틀려진다.

즉, 15개의 layer가 생성될 때, 비록 중요도는 낮지만 크기가 큰 layered stream이 생성되어질 수 있다. 이러한 stream은 시스템에 의해 네트워크 상황에 따라서 전송여부가 결정되지만, 결국 이 stream을 가지고 있는 노드의 부하는 커지게 된다. 따라서 향상된 형태의 layering 알고리즘과 시스템의 개발이 필요하다.

기존 system에서 하나의 MPEG 비디오 데이터에 대하여 layering을 수행한 후에는 항상 15개의 고정된 형태의 layered stream이 생성되었다. 즉, 아무리 크기가 작은 stream이라고 해도 15개 stream으로, 큰 stream이라고 해도 15개의 stream으로 나누어진 것이다.

그러나, frame의 크기가 작거나, frame rate가 작아서 초당 전송될 크기가 작은 stream에 대해서는 굳이 15개의 세분화된 형태의 stream이 필요하지 않아도 될 경우가 있다. 반대로 큰 stream의 경우에, 좀더 세분화된 형태의 layering 시스템이 필요한 경우도 있을 것이다. 즉, 15개의 고정된 형태의 layering이 아닌, 상황에 따라 다양한 개수의 layering stream으로 나누는 것이 필요하다.

5. 결론

본 논문은 가변적인 네트워크 환경과 같은 멀티미디어 전송 환경에 따른 문제점을 해결하기 위해 필요한 방법인 layering 기법에 대해 알아보았고, 그 기법을 이용하여 병렬 VOD서버에 보다 효율적으로 stream을 분산 저장하는 방법에 대해서도 연구해 보았다.

논문을 통해 layered stream들 중에서 기본계층에 가까운 stream일수록 접근 회수가 증가하는 것을 dependency tree를 통해서 규명하였다. 그리고 네트워크 환경에 따른 layering stream에 대한 접근 pattern을 알 수가 있었다. 이러한 자료를 토대로 생성되어진 layered stream의 배치는 서버측에 layered stream의 저

장을 수행할 때, load-balancing 효과를 줄 수 있었다.

[참고문헌]

- [1] 김태영, "Space Striping Policy를 이용한 TOF(Temporal qOs Filtering) 기법", 석사학위논문, 충남대학교 컴퓨터과학과, 2000
- [2] 손호신, "Scalable Video-on-Demand를 위한 TFS(Temporal-Fidelity Scaling) 기법", 석사학위논문, 충남대학교 컴퓨터과학과, 2000
- [3] 정제창, "그림으로 보는 최신 MPEG", 교보문고, 1995
- [4] International Standard ISO/IEC 14496-2, Information technology coding of audio-visual object-part 2: Visual, 1999